IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.:	10/790,966	Conf. No.:	2143
Inventor:	Gaur et al.		
Filed:	March 2, 2004		
TC/AU:	2132		
Examiner:	Cordelia P. Kane		
Docket No.:	RPS920020014US1 (IRA-10-6345)		
Customer No.:	26675		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

APPEAL BRIEF

Dear Sir:

Attached herewith is an Appeal Brief pursuant to 35 U.S.C. §134 and 37 C.F.R. §41.37 for the above-identified patent application in support of a Notice of Appeal filed at the US Patent and Trademark Office on January 23, 2009.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest in the above-entitled application is International Business Machines Corporation, Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

The undersigned attorney/agent, the appellants, and the assignee are not aware of any related appeals or interferences that would directly affect, or be directly affected by, or have a bearing on the Board's decision in this pending appeal.

III. STATUS OF THE CLAIMS

Claims 1-26 are pending, and are all on appeal. Claims 1-26 stand rejected.

IV. STATUS OF AMENDMENTS

No after final amendments have been submitted.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Independent **claim 1** is directed towards a system for secure data transfer over a network (*See, inter alia,* p. 4, lines 13-14; Abstract; and Fig. 1), the system including memory (*See, inter alia,* p. 4, lines 25-28; Abstract; and Fig. 1); a network interface coupled to the memory controller (*See, inter alia,* p. 7, lines 23-24; and Fig 1), the network interface comprising: a first data moving unit (DMU) configured to exchange secure data with a first portion of the network (*See, inter alia,* p. 7, lines 24-26; and Fig. 1); a second DMU configured to exchange non-secure data with a second portion of the network (*See, inter alia,* p. 7, lines 27-29; and Fig. 1); and a processor coupled to the memory controller (*See, inter alia,* p. 6, lines 7-8; Abstract; and Fig. 1), the processor including: logic configured to identify information flow of the data in the memory (*See, inter alia,* p. 9, lines 10-21); logic configured to identify a priority of the identified information flow (*See, inter alia,* p. 9, lines

22-23); logic configured to retrieve a portion of the data from the memory using the memory controller based on the identified priority (*See, inter alia,* p. 9, lines 23-29); logic configured to perform security operations on the retrieved portion of the data (*See, inter alia,* p. 6, lines 11-17); logic configured to store the operated-on portion of the data in the memory using the memory controller (*See, inter alia,* p. 7, lines 14-17); logic configured to queue data for transfer based on the identified priority (*See, inter alia,* p. 9, lines 23-26) and logic configured to discard portions of data associated with a particular information flow based on the identified priority (*See, inter alia,* p. 10, lines 8-12; and p. 11, line 29 to p. 12, line 3); wherein the memory controller is further configured to transfer the operated-on portion of the data from the memory to the network (*See, inter alia,* p. 7, lines 17-18; and Abstract), wherein portions of the data having higher priority information flow are retrieved before portions of the data having lower priority information flow based on the identified priority (*See, inter alia,* p. 9, line 30 to p. 10, line 2), wherein the priority of information flow is independent of an order in which the data is stored in the memory and any contentions for memory (*See, inter alia,* p. 5, lines 1-6; and p. 9, lines 22-29).

Independent **claim 12** is directed towards a method for secure data transfer over a network (*See, inter alia,* p. 11, lines 4-5; Abstract; and Fig. 2), the method comprising: transferring data from the network to memory using a memory controller (*See, inter alia,* p. 11, lines 6-7; Abstract; and Fig. 2); identifying information flow of the data in the memory (*See, inter alia,* p. 11, lines 23-26); identifying a priority of the identified information flow (*See, inter alia,* p. 11, lines 26); retrieving a portion of the data from the memory based on the identified priority into a processor using the memory controller (*See, inter alia,* p. 11, lines 26-29), wherein portions of the data having higher priority information flow are retrieved before portions of the data having lower priority information flow (*See, inter alia,* p. 9 line 30 to p. 10, line 2), wherein the priority of information flow is independent of an order in which the data is stored in the memory and any memory contentions (*See, inter alia,* p. 5, lines 1-6; and p. 9, lines 22-29); performing security operations on the retrieved portion of the data

using the processor (*See, inter alia,* p. 11, lines 9-10; and Fig. 2); storing the operated-on portion of the data in the memory using the memory controller (*See, inter alia,* p. 11, lines 13-14; and Fig. 2); discarding portions of data associated with particular information flow based on the identified memory (*See, inter alia,* p. 11, line 29 to p. 12, line 3); queuing the operated-on portion of the data for transfer based on the identified priority (*See, inter alia,* p. 11, line 29 to p. 12, line 3); and transferring the operated-on portion of the data from the memory to the network using the memory controller (*See, inter alia,* p. 11, lines 14 -16; and Fig. 2).

Independent **claim 22** is directed towards a computer readable storage medium containing a computer program for secure data transfer over a network, wherein the computer program comprises executable instructions for (*See, inter alia,* p. 12, lines 26-31): transferring data from the network to memory using a memory controller (*See, inter alia,* p. 11, lines 6-7; Abstract; and Fig. 2); identifying information flow of the data in the memory; identifying a priority of the identified information flow (*See, inter alia,* p. 11, lines 23-26); retrieving a portion of the data from the memory into a processor using the memory controller based on the identified priority (*See, inter alia,* p. 11, lines 26-29); performing security operations on the retrieved portion of the data using the processor (*See, inter alia,* p. 11, lines 9-10; and Fig. 2); storing the operated-on portion of the data in the memory using the memory controller (*See, inter alia,* p. 11, lines 13-14; and Fig. 2); discarding portions of data associated with particular information flow based on the identified memory (*See, inter alia,* p. 11, line 29 to p. 12, line 3); queuing the operated-on portion of the data for transfer based on the identified priority (*See, inter alia,* p. 11, line 29 to p. 12, line 3); and transferring the operated-on portion of the data from the memory to the network using the memory controller (*See, inter alia,* p. 11, lines 14 -16; and Fig. 2), wherein operated-on portions of the data having higher priority information flow are transferred before portions of the data having lower priority information flow (*See, inter alia,* p. 9, line 30 to p. 10, line 2), wherein the priority does not depend on a location of the operated-on data in the memory and any memory contention (*See, inter alia,* p. 5, lines 1-6; and p. 9, lines 22-29).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1, 2, 4-7, 9, 12-16 and 20-25 are unpatentable under 35 U.S.C. 103(a) over Noehring et al. (US 2002/0188871) in view of Ganesan et al. (US 2003/0069973).

Whether claim 3 is unpatentable under 35 U.S.C. 103(a) over Noehring et al. in view of Ganesan et al., and further in view of Kocaman et al. (US 2004/0030513).

Whether claims 8, 17 and 26 are unpatentable under 35 U.S.C. 103(a) over Noehring et al. in view of Ganesan et al., and further in view of Nozawa et al. (US 5,235,641).

Whether claims 10, 11, 18 and 19 are unpatentable under 35 U.S.C. 103(a) over Noehring et al. in view of Ganesan et al., and further in view of Trost et al. (US 4,627,018).

VII. ARGUMENTS

A. The Rejection of Claims 1, 2, 4-7, 9, 12-16 and 20-25 under 35 U.S.C. 103(a)

Claims 1, 2, 4-7, 9, 12-16 and 20-25 stand rejected under 35 U.S.C. 103(a) as being unpatentable over 35 U.S.C. 103(a) over Noehring et al. in view of Ganesan et al. This rejection should be reversed because the combination of the cited references fails to establish a *prima facie* case of obviousness with respect to the subject claims.

> The rationale to support a conclusion that the claim would have been obvious is that all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed. *KSR International Co. v. Teleflex Inc.*, 550 U.S. ___ (2007). MPEP §2143.

Independent **claim 1** is directed to a system for secure data transfer over a network including, *inter alia*, logic configured to discard portions of data associated with a particular information flow based on the identified priority. The combination of Noehring et al. and Ganesan et al. does not teach or suggest this claim aspect.

The Office concedes that Noehring et al. does not disclose this claim aspect. In an attempt to make up for this conceded deficiency, the Office asserts that Ganesan et al. discloses receiving a packet and determining which class it belongs to, and possibly dropping the packet based on the class it belongs to (page 11, paragraph 128). The Office concludes that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the memory access of Noehring et al. to include the classification and prioritization of Ganesan et al. Appellants respectfully disagree as Ganesan et al. fails to teach the conceded deficiencies of Noehring et al.

Particularly, the cited section of Ganesan et al. fails to teach discarding portions of data associated with a particular information flow based on the identified priority. Instead, the cited section of Ganesan et al. teaches that when a packet arrives in a multiprocessing system, it is first classified to determine in which traffic class it belongs. Once this classification has been made, the packet is placed in a queue along with other packets of the same class (¶ [0128]). The scheduler chooses packets for transmission from the queues in such a way that the relative bandwidth allocation among the queues is maintained (¶ [0128]). If packets for a given class arrive faster than they can be drained from the respective queue (i.e. the class is consuming more bandwidth than has been allocated for it) the queue depth will increase and the senders of that traffic class must be informed to lower their transmission rates before the queue completely overflows (¶ [0128]). As such, Ganesan et al. is silent with respect to discarding portions of data associated with a particular information flow based on the identified priority. In other words, Ganesan et al. fails to teach discarding packets from the queues of any traffic class as required by claim 1.

Accordingly, this rejection should be reversed.

Independent **claims 12 and 22** recite claim aspects similar to those recited in claim 1. As such, the above discussion with respect to claim 1 applies *mutatis mutandis* to claims 12 and 22, and this rejection should be reversed.

Claims 2, 4-7, 9, 13-16, 20-21 and 23-25 depend respectively from independent claims 1, 12 and 22, and are allowable at least by virtue of their dependencies.

B. The Rejection of Claim 3 under 35 U.S.C. 103(a)

Claim 3 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Noehring et al. in view of Ganesan et al., and further in view of Kocaman et al. **Claim 3** indirectly depends from independent claim 1 and is allowable at least by virtue of this dependency.

C. The Rejection of Claims 8, 17 and 26 under 35 U.S.C. 103(a)

Claims 8, 17 and 26 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Noehring et al. in view of Ganesan et al., and further in view of Nozawa et al. **Claims 8, 17 and 26** respectively depend from independent claims 1, 12 and 22 and are allowable at least by virtue of their dependencies.

D. The Rejection of Claims 10, 11, 18 and 19 under 35 U.S.C. 103(a)

Claims 10, 11, 18 and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Noehring et al. in view of Ganesan et al., and further in view of Trost et al. **Claims 10, 11, 18 and 19** respectively depend directly or indirectly from independent claims 1 and 12 and are allowable at least by virtue of their dependencies.

CONCLUSION

In view of the foregoing, it is submitted that the claims distinguish patentably and non-obviously over the prior art of record, and reversal of the rejection of the claims herein is respectfully requested.

Respectfully submitted,

Driggs, Hogg, Daugherty & Del Zoppo Co., L.P.A.

March 20, 2009

/Michael J. Corrigan/
Michael J. Corrigan, Reg. No. 42,440
CUSTOMER NO. 26675

MJC:cg

VIII. CLAIM APPENDIX

1. (Previously presented) A system for secure data transfer over a network, the system comprising:

memory;

a memory controller configured to transfer data received from the network to the memory;

a network interface coupled to the memory controller, the network interface comprising:

a first data moving unit (DMU) configured to exchange secure data with a first portion of the network;

a second DMU configured to exchange non-secure data with a second portion of the network; and

a processor coupled to the memory controller, the processor including:

logic configured to identify information flow of the data in the memory;

logic configured to identify a priority of the identified information flow;

logic configured to retrieve a portion of the data from the memory using the memory controller based on the identified priority;

logic configured to perform security operations on the retrieved portion of the data;

logic configured to store the operated-on portion of the data in the memory using the memory controller;

logic configured to queue data for transfer based on the identified priority and logic configured to discard portions of data associated with a particular information flow based on the identified priority;

wherein the memory controller is further configured to transfer the operated-on portion of the data from the memory to the network, wherein portions of the data having higher priority information flow are retrieved before portions of the data having lower priority

information flow based on the identified priority, wherein the priority of information flow is independent of an order in which the data is stored in the memory and any contentions for memory.

2. (Previously presented) The system of claim 1, wherein the first and second DMUs directly communicate with the first and seconds portions of the network.

3. (Original) The system of claim 2, wherein the network interface comprises:

a first serializer/deserializer (SERDES) circuit coupled between the first DMU and the first network portion and a second SERDES coupled between the second DMU and the second network portion, each SERDES configured to convert serial data received from the respective network portions to a parallel format and to convert parallel data received from the respective DMUs to a serial format.

4. (Original) The system of claim 1, wherein the logic configured to perform security operations comprises:

logic configured to obscure the portion of the data when the retrieved portion is non-secure data;

logic configured to decipher the portion of the data when the retrieved portion is secure data; and

logic configured to determine an integrity of the portion of data.

5. (Original) The system of claim 1, wherein the processor comprises:

logic configured to perform quality-of-service (QoS) operations on the data in coordination with performing the security operations.

6. (Previously presented) The system of claim 5, wherein the logic configured to perform QoS operations comprises:

logic configured to identify an information flow associated with the portion of the data;

logic configured to determine a priority of the information flow; and

logic configured to schedule at least one of the retrieving the portion of the data and the transferring of the operated-on portion of the data from memory based on the priority of the information flow associated with the portion of the data.

7. (Original) The system of claim 6, wherein the processor comprises:

logic configured to decipher the portion of the data prior to the identifying of the information flow when the retrieved portion is secure data; and

logic configured to obscure the portion of the data after the identifying of the information flow when the retrieved portion is non-secure data.

8. (Original) The system of claim 1, wherein the processor comprises:

logic configured to compress the portion of the data using the processor prior to performing the security operations when the retrieved portion is non-secure data; and

logic configured to decompress the portion of the data in the processor after performing the security operations when the retrieved portion is secure data.

9. (Original) The system of claim 1, wherein the memory includes a memory block having a plurality of memory banks, the memory controller comprising:

logic configured to reference the plurality of memory banks in a sequence that minimizes a memory access time.

10. (Original) The system of claim 1, wherein the memory controller comprises:

logic configured to include a request to reference the memory into one of a group of read requests and a group of write requests; and

logic configured to execute all requests included in one of the groups of read requests and write requests before executing a request included in the other group.

11. (Original) The system of claim 10, comprising:

logic configured to include error correction code with the data transferred to or stored in the memory; and

logic configured to detect and correct errors in the data retrieved or transferred from the memory based on the error correction code included with the data.

12. (Previously presented) A method for secure data transfer over a network, the method comprising:

transferring data from the network to memory using a memory controller;

identifying information flow of the data in the memory;

identifying a priority of the identified information flow;

retrieving a portion of the data from the memory based on the identified priority into a processor using the memory controller, wherein portions of the data having higher priority information flow are retrieved before portions of the data having lower priority information flow, wherein the priority of information flow is independent of an order in which the data is stored in the memory and any memory contentions;

performing security operations on the retrieved portion of the data using the processor;

storing the operated-on portion of the data in the memory using the memory controller;

discarding portions of data associated with particular information flow based on the identified memory;

queuing the operated-on portion of the data for transfer based on the identified priority; and

transferring the operated-on portion of the data from the memory to the network using the memory controller.

13. (Original) The method of claim 12, wherein the security operations comprise at least one of:

obscuring the portion of the data when the retrieved portion is non-secure data;

deciphering the portion of the data when the retrieved portion is secure data; and

determining an integrity of the portion of data.

14. (Original) The method of claim 12, comprising:

performing quality-of-service (QoS) operations on the data in coordination with performing the security operations using the processor.

15. (Original) The method of claim 14, wherein the QoS operations comprise:

identifying an information flow associated with the portion of the data;

determining a priority of the information flow; and

scheduling at least one of the retrieving the portion of the data and the transferring the operated-on portion of the data from memory based on the priority of the information flow associated with the portion of the data.

16. (Original) The method of claim 15, comprising:

deciphering the portion of the data prior to the identifying of the information flow when the retrieved portion is secure data; and

obscuring the portion of the data after the identifying of the information flow when the retrieved portion is non-secure data.

17. (Original) The method of claim 12, comprising:

compressing the portion of the data using the processor prior to performing the security operations when the retrieved portion is non-secure data; and

decompressing the portion of the data in the processor after performing the security operations when the retrieved portion is secure data.

18. (Original) The method of claim 12, comprising:

including a request to reference the memory into one of a group of read requests and a group of write requests; and

executing all requests included in one of the groups of read requests and write requests before executing a request included in the other group.

19. (Original) The method of claim 18, wherein the executing all requests included in one of the groups of read requests and write requests occurs when a sum of the requests included in one of the groups corresponds to a predetermined amount of the memory.

20. (Original) The method of claim 12, comprising:

including error correction code with the data transferred to or stored in the memory; and

at least one of detecting and correcting errors in the data retrieved or transferred from the memory based on the error correction code included with the data.

21. (Original) The method of claim 12, comprising:

referencing portions of the memory in a sequence that minimizes a memory access time.

22. (Previously presented) A computer readable storage medium containing a computer program for secure data transfer over a network, wherein the computer program comprises executable instructions for:

transferring data from the network to memory using a memory controller;

identifying information flow of the data in the memory;

identifying a priority of the identified information flow;

retrieving a portion of the data from the memory into a processor using the memory controller based on the identified priority;

performing security operations on the retrieved portion of the data using the processor;

storing the operated-on portion of the data in the memory using the memory controller;

discarding portions of data associated with particular information flow based on the identified memory;

queuing the operated-on portion of the data for transfer based on the identified priority; and

transferring the operated-on portion of the data from the memory to the network using the memory controller, wherein operated-on portions of the data having higher priority information flow are transferred before portions of the data having lower priority information flow, wherein the priority does not depend on a location of the operated-on data in the memory and any memory contention.

23. (Previously presented) The computer readable storage medium of claim 22, wherein the computer program comprises executable instructions for:

obscuring the portion of the data when the retrieved portion is non-secure data;

deciphering the portion of the data when the retrieved portion is secure data; and

determining an integrity of the portion of data.

24. (Previously presented) The computer readable storage medium of claim 22, wherein the computer program comprises executable instructions for:

performing quality-of-service (QoS) operations on the data in coordination with performing the security operations using the processor.

25. (Previously presented) The computer readable storage medium of claim 24, wherein the computer program comprises executable instructions for:

identifying an information flow associated with the portion of the data;

determining a priority of the information flow; and

scheduling at least one of the retrieving the portion of the data and the transferring the operated-on portion of the data from memory based on the priority of the information flow associated with the portion of the data.

26. (Previously presented) The computer readable storage medium of claim 22, wherein the computer program comprises executable instructions for:

compressing the portion of the data using the processor prior to performing the security operations when the retrieved portion is non-secure data; and

decompressing the portion of the data in the processor after performing the security operations when the retrieved portion is secure data.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None known to undersigned attorney/agent.